Cross-validation improved by aggregation: Agghoo

Guillaume Maillard, Sylvain Arlot, Matthieu Lerasle

Cross-validation improved by aggregation: Agghoo

Guillaume Maillard
Ecole Normale Supérieure
Paris 75005, France
guillaume.maillard@ens.fr

Sylvain Arlot
Laboratoire de Mathématiques d'Orsay,
Univ. Paris-Sud, CNRS, Université Paris-Saclay,
91405 Orsay, France
sylvain.arlot@math.u-psud.fr

Matthieu Lerasle
Laboratoire de Mathématiques d'Orsay,
Univ. Paris-Sud, CNRS, Université Paris-Saclay,
91405 Orsay, France
matthieu.lerasle@math.u-psud.fr

September 11, 2017

Abstract

Cross-validation is widely used for selecting among a family of learning rules. This paper studies a related method, called aggregated hold-out (Agghoo), which mixes cross-validation with aggregation; Agghoo can also be related to bagging. According to numerical experiments, Agghoo can improve significantly cross-validation's prediction error, at the same computational cost; this makes it very promising as a general-purpose tool for prediction. We provide the first theoretical guarantees on Agghoo, in the supervised classification setting, ensuring that one can use it safely: at worse, Agghoo performs like the hold-out, up to a constant factor. We also prove a non-asymptotic oracle inequality, in binary classification under the margin condition, which is sharp enough to get (fast) minimax rates.

1 Introduction

Machine learning rules almost always depend on some hyperparameters, whose choice has a strong impact on the final performance. For instance, nearest-

neighbor rules [Biau and Devroye, 2015] depend on the number k of neighbors and on some distance measure over the feature space. Kernel methods [Scholkopf and Smola, 2001] require to choose an appropriate kernel. A third example, among many others, is given by regularized empirical risk minimization rules, such as support vector machines [Steinwart and Christmann, 2008] or the Lasso [Tibshirani, 1996, Bühlmann and van de Geer, 2011], which all depend on some regularization parameter. More generally, the problem of choosing from data among a family of learning rules is central to machine learning, including model selection [Burnham and Anderson, 2002, Massart, 2007] and when one hesitates between different kinds of rules —for instance, support vector machines or random forests.

In supervised learning, cross-validation (CV) is a general, efficient and classical answer to this problem [Arlot and Celisse, 2010]. It relies on the idea of splitting data into a training sample —used for training a predictor with each rule in competition— and a validation sample —used for assessing the performance of each predictor. This leads to an estimator of the risk —the hold-out estimator when data are split once, the CV estimator when an average is taken over several data splits—, which can be minimized for selecting among a family of competing rules.

A completely different strategy, called aggregation, is to *combine* the predictors obtained with all candidate rules [Nemirovski, 2000, Yang, 2001, Tsybakov, 2004]. A major interest of aggregation is that it builds a learning rule in a much larger set that the family of rules in competition; therefore, it can sometimes yield a better performance than the best of all rules. Aggregation is the keystone of ensemble methods [Dietterich, 2000], among which we can mention bagging [Breiman, 1996], AdaBoost [Freund and Schapire, 1997] and random forests [Breiman, 2001, Biau and Scornet, 2016].

This paper studies a mix of cross-validation and aggregation ideas, called *aggregated hold-out* (Agghoo). Data are split several times; for each split, the hold-out selects one predictor; then, the predictors obtained with the different splits are aggregated. This procedure is as general as cross-validation and it has the same computational cost. Moreover, it seems to be folklore knowledge among practicioners that Agghoo (or its variants) performs better than CV for prediction. Yet, Agghoo has never been properly studied in the literature, even experimentally, to the best of our knowledge. The closest results we found [Jung and Hu, 2015, Jung, 2016] study other procedures, called ACV and EKCV, and prove weaker theoretical results than ours; we explain in Section 2.3 why Agghoo is more natural and should be preferred to ACV and EKCV in general.

Because of the aggregation step, Agghoo is an ensemble method, and it is particularly close to subagging —a variant of bagging where the bootstrap is replaced by subsampling— [Bühlmann and Yu, 2002], since both combine subsampling with aggregation. However, Agghoo is *not* subagging applied to the hold-out selected predictor, in which the subsample itself is split into training and validation samples; see Section 2.3. Therefore, we cannot apply previous re-

sults on subagging for analyzing Agghoo; new developments are required. Since bagging and subagging are well-known for their stabilizing effects [Breiman, 1996, Bühlmann and Yu, 2002], we can expect Agghoo to behave similarly; in particular, it should improve much the prediction performance of CV when the hold-out selected predictor is unstable. The simulation experiments of Section 4 confirm this intuition.

Contributions The purpose of this paper is to investigate both theoretical and practical performances of Agghoo. As a first step, we focus on supervised classification with the 0–1 loss. First, Section 3 provides three theoretical results on the classification error of Agghoo. Proposition 3.1 is a general result —valid for any family of learning rules— proving that the excess risk of Agghoo is smaller than the excess risk of hold-out selection, multiplied by the number of classes. This ensures that Agghoo can be used safely: one cannot loose too much by preferring Agghoo to the hold-out. We then focus on binary classification under the margin condition [Mammen and Tsybakov, 1999], which is well-known for allowing fast learning rates [Audibert and Tsybakov, 2007, Lecué, 2007, Audibert, 2009]. Theorem 3.2 is a non-asymptotic oracle inequality for Agghoo, showing that Agghoo performs almost as well as the best possible selection rule (called the oracle). We illustrate the sharpness of this oracle inequality by considering the setting of Audibert and Tsybakov [2007, Section 3]: Theorem 3.4 shows that Agghoo applied to a well-chosen family of classifiers yields a minimax-adaptive procedure —hence, optimal in worst-case. Finally, Section 4 is a numerical study of the performance of Agghoo, on synthetic and real data sets, with two different kinds of selection problems. It illustrates that Agghoo actually performs much better than hold-out, and even better than CV —provided its parameters are well-chosen. When choosing among decision trees, the prediction performance of Agghoo is much better than the one of CV —for the same computational cost—, which illustrates the strong interest of using Agghoo when the choice among competing learning rules is "unstable". Based upon our experiments, we also give in Section 4 some guidelines for choosing Agghoo's parameters: the training set size and the number of data splits.

2 Setting and definitions

2.1 Supervised classification

Consider the supervised classification problem where a sample (X_1, Y_1), ..., $(X_n, Y_n) \in \mathcal{X} \times \mathcal{Y}$ is given, with $\mathcal{X}$ any measurable space and $\mathcal{Y} = \{0, \dots, M-1\}$ for some integer $M \geq 2$. The goal is to build a classifier —that is, a measurable map $f : \mathcal{X} \to \mathcal{Y}$— such that, for any new observation (X, Y), $f(X)$ is a good prediction for Y. Throughout the paper, $(X_1, Y_1), \dots, (X_n, Y_n), (X, Y)$ are assumed independent with the same distribution P. Let $\mathcal{F}$ denote the set

of classifiers. The quality of any classifier $f \in \mathcal{F}$ is measured by its risk

$$R(f) = \mathbb{P}_{(X,Y)\sim P}\big(Y \neq f(X)\big)$$

and any optimal classifier $f^* \in \operatorname{argmin}_{f\in\mathcal{F}} R(f)$ is called a Bayes classifier [Devroye et al., 1996]. We also define the excess risk of $f \in \mathcal{F}$ by $\ell(f^*, f) = R(f) - R(f^*) = R(f) - \inf_{f\in\mathcal{F}} R(f) \geq 0$.

Since we only have access to P through the data $D_n := (X_i, Y_i)_{1\leq i\leq n}$, a classifier is built from data, thanks to a *classification rule* $G : \cup_{k\geq 1}(\mathcal{X}\times\mathcal{Y})^k \to \mathcal{F}$, which maps a sample (of any size) into a classifier.

2.2 Selection of classification rules by cross-validation

A generic situation is when a family $\mathcal{G}$ of classification rules is given, and not only a single rule, so that we have to select one of them —or to combine their outputs. For instance, we can consider the family $(G_k^{\mathrm{NN}})_{k\geq 1}$ of nearest neighbors classifiers when $\mathcal{X}$ is a metric space —the parameter k denoting the number of neighbors—, or the family $(G_\lambda^{\mathrm{SVM}})_{\lambda\in[0,+\infty)}$ of support vector machine classifiers for a given kernel on $\mathcal{X}$ —λ denoting the regularization parameter.

A common answer to this problem is to *select* one of these rules from data, and cross-validation methods are a general tool for doing so. Let us briefly recall these methods here; we refer the reader to the survey by Arlot and Celisse [2010] for details and references, and to the paper by Arlot and Lerasle [2016] for the most recent results.

For any non-empty $B \subset \{1,\dots,n\}$, define the corresponding sample and empirical risk by

$$D_n^B := (X_i, Y_i)_{i\in B} \qquad \text{and} \qquad \forall f \in \mathcal{F}, \qquad \widehat{\mathcal{R}}_B(f) := \frac{1}{|B|}\sum_{i\in B} \mathbb{1}_{\{f(X_i)\neq Y_i\}}\,,$$

respectively. Then, for any classification rule G, $\widehat{f}_{G,B} := G(D_n^B)$ is the (random) classifier obtained by training G on the subsample of D_n indexed by B.

Given a non-empty subset T of $\{1,\dots,n\}$ and some classification rule G, the hold-out estimator of the risk of G is defined by $\mathrm{HO}_T(G) := \widehat{\mathcal{R}}_{T^c}(\widehat{f}_{G,T})$. A classifier $\widehat{f}_{G,T}$ is built from the *training sample* D_n^T, then its quality is assessed on the *validation sample* $D_n^{T^c}$. Note that $\mathrm{HO}_T(G)$ depends on the sample D_n even if this does not appear in the notation. Then, hold-out can be used for *selecting* one rule among $\mathcal{G}$, by minimizing $\mathrm{HO}_T(G)$ over $G \in \mathcal{G}$:

$$\widehat{f}^{\,\mathrm{ho}}_{\mathcal{G},T} := \widehat{G}^{\mathrm{ho}}_{\mathcal{G},T}(D_n^T), \qquad \text{where} \qquad \widehat{G}^{\mathrm{ho}}_{\mathcal{G},T} \in \operatorname*{argmin}_{G\in\mathcal{G}} \mathrm{HO}_T(G) \quad . \tag{1}$$

We call $\widehat{f}^{\,\mathrm{ho}}_{\mathcal{G},T}$ the *hold-out classifier*.

Hold-out depends on the arbitrary choice of a training set T, and is known to be quite unstable, despite its good theoretical properties [Massart, 2007, Section 8.5.1]. Therefore, practicioners often prefer to use cross-validation instead,

which considers several training sets. Given a family $\mathcal{T} = \{T_1, \ldots, T_V\}$ of training sets, the cross-validation risk estimator of any classification rule G is defined by

$$\mathrm{CV}_{\mathcal{T}}(G) = \frac{1}{V} \sum_{j=1}^{V} \mathrm{HO}_{T_j}(G) \quad ,$$

leading to the *cross-validation classifier*

$$\widehat{f}^{\mathrm{cv}}_{\mathcal{G},\mathcal{T}} := \widehat{G}^{\mathrm{cv}}_{\mathcal{G},\mathcal{T}}(D_n) \qquad \text{where} \qquad \widehat{G}^{\mathrm{cv}}_{\mathcal{G},\mathcal{T}} \in \operatorname*{argmin}_{G \in \mathcal{G}} \mathrm{CV}_{\mathcal{T}}(G) \ . \tag{2}$$

Depending on how $\mathcal{T}$ is chosen, this can lead to leave-one-out, leave-p-out, V-fold cross-validation or Monte-Carlo cross-validation, among others [Arlot and Celisse, 2010].

2.3 Aggregated hold-out (Agghoo)

In this paper, we study another way to improve on the stability of hold-out classifiers, by *aggregating* the hold-out classifiers $\widehat{f}^{\mathrm{ho}}_{\mathcal{G},T_j}$ obtained from several training sets $T_1, \ldots, T_V$. Formally, the *aggregated hold-out classifer* (Agghoo) is obtained by making a *majority vote* among them:

$$\forall x \in \mathcal{X}, \qquad \widehat{f}^{\mathrm{ag}}_{\mathcal{G},\mathcal{T}}(x) \in \operatorname*{argmax}_{y \in \mathcal{Y}} \frac{1}{V} \sum_{j=1}^{V} \mathbb{1}_{\left\{\widehat{f}^{\mathrm{ho}}_{\mathcal{G},T_j}(x) = y\right\}} \ . \tag{3}$$

Compared to cross-validation classifiers defined by Eq. (2), we reverse the order between aggregation (majority vote or averaging) and minimization of the risk estimator. To the best of our knowledge, Agghoo has not appeared before in the literature. The closest procedures we found are "K-fold averaging cross-validation" (ACV) proposed by Jung and Hu [2015] for linear regression, and "efficient K-fold cross-validation" (EKCV) proposed by Jung [2016] for high-dimensional regression. The main difference with Agghoo is that ACV and EKCV average the chosen *parameters* —the models for ACV, the regularization parameters for EKCV—, whereas Agghoo averages the chosen *classifiers*. This leads to completely different procedures for learning rules that are not linear functions of their parameters. Even in the case of linear regression with ACV, where the estimator is a linear function of the projection matrix onto the model —which is the "parameter" averaged by ACV—, Agghoo would lead to a different procedure since it averages the $\widehat{G}^{\mathrm{ho}}_{\mathcal{G},T_j}(D_n^{T_j})$ while ACV averages the $\widehat{G}^{\mathrm{ho}}_{\mathcal{G},T_j}(D_n)$. In the general case, and in particular for classification, averaging the classifiers $\widehat{G}^{\mathrm{ho}}_{\mathcal{G},T_j}(D_n^{T_j})$, which have been selected for their good performance on the validation set T_j^c, is more natural than averaging the $\widehat{G}^{\mathrm{ho}}_{\mathcal{G},T_j}(D_n)$ whose performance has not been assessed on independent data.

Another related procedure —not explicitly studied in the literature— is subagging applied to hold-out selection $D_n \mapsto \widehat{G}^{\mathrm{ho}}_{\mathcal{G},T}(D_n^T)$, as defined by Eq. (1).

Denoting by $T_1, \dots, T_V$ the subsample sets chosen by the subagging step, and by T'_j the training set used by the hold-out applied to the subsample $D_n^{T_j}$, the subagged hold-out predictor is obtained by making a majority vote among $\widehat{f}_{\widehat{G}^{\mathrm{ho}}_{\mathcal{G},T'_j}(D_n^{T_j})}(D_n^{T_j}) = \widehat{G}^{\mathrm{ho}}_{\mathcal{G},T'_j}(D_n^{T_j})$. Compared to Agghoo, the main difference is that this aggregates predictors trained on a *subsample* of $D_n^{T_j}$ —instead of the entire $D_n^{T_j}$.

3 Theoretical guarantees

In this section, we make the following two assumptions on the training sets:

$$T_1, \dots, T_V \subset \{1, \dots, n\} \text{ are independent of } D_n = (X_i, Y_i)_{1 \le i \le n}\,, \tag{H1}$$

$$|T_1| = \cdots = |T_V| = n - p \in \{1, \dots, n-1\}\,. \tag{H2}$$

These assumptions are satisfied for the most classical cross-validation methods, including leave-p-out, V-fold cross-validation (with $p = n/V$) and Monte-Carlo cross-validation [Arlot and Celisse, 2010].

General bound Our first theoretical result connects the performances of Agghoo and hold-out.

Proposition 3.1 *Let $\mathcal{G}$ denote a collection of classification rules and $\mathcal{T}$ a collection of training sets satisfying* (H1)–(H2)*. The aggregated hold-out estimator $\widehat{f}^{\mathrm{ag}}_{\mathcal{G},\mathcal{T}}$ defined by Eq.* (3) *and the hold-out estimator defined by Eq.* (1) *satisfy:*

$$\mathbb{E}\big[\ell(f^*, \widehat{f}^{\mathrm{ag}}_{\mathcal{G},\mathcal{T}})\big] \le M\mathbb{E}\big[\ell(f^*, \widehat{f}^{\mathrm{ho}}_{\mathcal{G},T_1})\big] \qquad \textit{and} \qquad \mathbb{E}\big[R(\widehat{f}^{\mathrm{ag}}_{\mathcal{G},\mathcal{T}})\big] \le 2\mathbb{E}\big[R(\widehat{f}^{\mathrm{ho}}_{\mathcal{G},T_1})\big]\ .$$

Proposition 3.1 is proved in supplementary material.

Up to the multiplicative constant M, the Agghoo predictor $\widehat{f}^{\mathrm{ag}}_{\mathcal{G},\mathcal{T}}$ performs theoretically at least as well as the hold-out selected estimator. When the number of classes M is small —for instance, in the binary case—, this guarantees that Agghoo's prediction error cannot be too large. Let us now use this general bound to obtain oracle and minimax properties for Agghoo.

Oracle inequality in binary classification From now on, we focus on the binary classification case, that is, $M = 2$. A classical assumption in this setting is the so-called *margin assumption* [Mammen and Tsybakov, 1999], for some $\beta \ge 0$ and $c \ge 1$:

$$\forall h > 0, \qquad \mathbb{P}\big(\big|2\eta(X) - 1\big| \le h\big) \le ch^\beta \qquad \text{where} \qquad \eta(X) := \mathbb{P}(Y = 1|X)\ ; \tag{MA}$$

η is called the regression function. Agghoo satisfies the following non-asymptotic oracle inequality under (MA).

Theorem 3.2 *Let $\mathcal{G}$ denote a collection of classification rules and $\mathcal{T}$ a collection of training sets satisfying* (H1)–(H2)*. If $\beta \geq 0$ and $c \geq 1$ exist such that* (MA) *holds true, we have:*

$$\mathbb{E}\big[\ell(f^*, \widehat{f}^{\mathrm{ag}}_{\mathcal{G},\mathcal{T}})\big] \leq 3\mathbb{E}\left[\inf_{G\in\mathcal{G}} \ell(f^*, \widehat{f}_{G,T_1})\right] + \frac{29c^{\frac{1}{\beta+2}}\log(e|\mathcal{G}|)}{p^{\frac{\beta+1}{\beta+2}}} \ .$$

Theorem 3.2 is proved in supplementary material. Note that the constant 3 in front of the oracle excess risk $\mathbb{E}[\inf_{G\in\mathcal{G}} \ell(f^*, \widehat{f}_{G,T_1})]$ can be made as close as desired to 2, at the price of enlarging the constant 29 in the remainder term.

Theorem 3.2 shows that Agghoo performs as well as the best classification rule in the collection $\mathcal{G}$ trained with $n-p$ data (called the oracle classifier), provided that the remainder term $\mathcal{O}(\log(e|\mathcal{G}|)/p^{\frac{\beta+1}{\beta+2}})$ can be neglected. This is a strong result, since it shows that Agghoo attains the same learning rates as the oracle classifier. In comparison, the results proved by Jung and Hu [2015] on ACV are much weaker, since they do not allow to derive any rate for ACV, except in the "parametric" setting.

Minimax rates We now address the main possible limitation of Theorem 3.2: is the remainder term negligible in front of the oracle excess risk in some "interesting" frameworks? To this end, we consider the setting of Audibert and Tsybakov [2007, Section 3], which is an example where fast minimax rates are known under the margin assumption. Let us briefly describe this setting. For any $\gamma, L > 0$, let $\Sigma_{\gamma,L}$ denote the class of functions with γ-Hölder semi-norm smaller than L (a formal definition is given in supplementary material). The classes of probability distributions that we'll be interested in is defined as follows.

Definition 3.3 *For any $\gamma, L > 0$, $\beta \geq 0$ and $c \geq 1$, $\mathcal{P}_{\gamma,L,\beta,c}$ denotes the set of probability distributions P on $\mathbb{R}^d \times \{0,1\}$ such that* (MA) *holds true —with parameters β, c—, the regression function $\eta \in \Sigma_{\gamma,L}(\mathbb{R}^d)$ and X has a density q with respect to the Lebesgue measure satisfying:*

$$\frac{1}{L}\mathbb{1}_{[0;1]^d} \leq q(X) \leq L\mathbb{1}_{[0;1]^d} \qquad \textit{almost surely.}$$

By Audibert and Tsybakov [2007, Theorem 3.5], the minimax rate of convergence over $\mathcal{P}_{\gamma,L,\beta,c}$ is $n^{-\frac{\gamma(1+\beta)}{2\gamma+d}}$ if $\gamma\beta \leq d$; this result is recalled in supplementary material. When $\gamma\beta < d$, we have $\gamma(1+\beta)/(2\gamma+d) < (\beta+1)/(\beta+2)$, so the remainder term in Theorem 3.2 can be neglected in front of the minimax rate of convergence over $\mathcal{P}_{\gamma,L,\beta,c}$, if $\mathcal{G}$ is a *polynomial collection* —that is, $|\mathcal{G}| \leq n^\alpha$ for some $\alpha \geq 0$— and if $p \geq \delta n$ with $\delta > 0$.

As detailed by Audibert and Tsybakov [2007, Section 3], having $\gamma\beta > d$ is a strong constraint on P, hence we can leave it aside in the following without losing too much. We can also remark that when $L = +\infty$ —which removes the condition on the density q of X—, Audibert and Tsybakov [2007, Section 4] show that the minimax rate then is $n^{-\frac{\gamma(1+\beta)}{\gamma(2+\beta)+d}}$ —even when $\gamma\beta \geq d$—, and this

rate is always larger than the remainder term in Theorem 3.2 if $\mathcal{G}$ and p are taken as previously.

Let us now explain how Agghoo can be used for building a minimax classifier, simultaneously over all classes $\mathcal{P}_{\gamma,L,\beta,c}$ such that $\gamma\beta < d$. The construction relies on a family of classification rules proposed by Audibert and Tsybakov [2007, Definition 2.3]. For any $\ell \geq 1$ and $h > 0$, $G^{\mathrm{LP}}_{\ell,h}$ is a plug-in classification rule based upon a local-polynomial Gaussian kernel estimator $\widehat{Q}_{\ell,h}$ of η, where ℓ is the polynomial degree and h is the kernel bandwidth. Then, $G^{\mathrm{LP}}_{\lfloor\gamma\rfloor,h_n(\gamma,d)}$ is minimax over $\mathcal{P}_{\gamma,L,\beta,c}$, where $h_n(\gamma,d)$ is a well-chosen bandwidth. This result, and the full definition of $G^{\mathrm{LP}}_{\ell,h}$, are recalled in supplementary material. Its main limitation is that the minimax rule depends on γ which is unknown; in other words, it is not *adaptive* to the smoothness γ of the regression function. The next theorem shows that Agghoo is a simple way to get rid of this issue.

Theorem 3.4 *Let $\tau \in (0,1)$ be fixed. For any $n \geq 1$, let $\mathcal{T}_n = \{T_1, \ldots, T_V\}$ be a collection of training sets satisfying* (H1)–(H2) *and $|T_1| = \lfloor \tau n \rfloor$, for some $V \geq 1$ (possibly depending on n), and let us define the collection*

$$\mathcal{G}^{\mathrm{LP}}_n := \left(G^{\mathrm{LP}}_{\ell,\frac{1}{k}}\right)_{1\leq\ell\leq n, 1\leq k\leq n}$$

of classification rules, where $G^{\mathrm{LP}}_{l,h}$ are defined above. Then, the Agghoo classifier $\widehat{f}^{\mathrm{ag}}_{\mathcal{G}^{\mathrm{LP}}_n,\mathcal{T}_n}$ defined by Eq. (3) *is minimax over classes $\mathcal{P}_{\gamma,L,\beta,c}$ simultaneously for all γ, L, β, c such that $\gamma\beta < d$.*

Theorem 3.4 is proved in supplementary material. It shows that Agghoo applied to a well-chosen collection of local polynomial estimators defined by Audibert and Tsybakov [2007] yields a *minimax adaptive* classifier. Let us emphasize that minimax adaptivity is a strong theoretical property.

Of course, Agghoo is not the only way to obtain such an adaptivity result —for instance, hold-out selection is sufficient to get it, as can be seen by taking $V = 1$ in Theorem 3.4—, even if we have not found such a minimax-adaptive statement for this problem in the literature. The main goal of Theorem 3.4 is to illustrate that our oracle inequality (Theorem 3.2, from which the minimax adaptivity of Agghoo derives) is sharp, and that Agghoo can lead to optimal classifiers, in one non-trivial setting at least.

4 Numerical experiments

This section illustrates the performance of Agghoo and CV on synthetic and real data.

4.1 k-nearest neighbors classification

We first consider the collection $\mathcal{G}^{\mathrm{NN}} = (G^{\mathrm{NN}}_k)_{k\geq 1,\, k \text{ odd}}$ of nearest-neighbors classifiers —assuming k is odd for avoiding ties— on the following binary classification problem.

Experimental setup Data $(X_1, Y_1), \dots, (X_n, Y_n)$ are independent, with X_i uniformly distributed over $\mathcal{X} = [0,1]^2$ and

$$\mathbb{P}(Y_i = 1 | X_i) = \sigma\left(\frac{g(X_i) - b}{\lambda}\right) \quad \text{where} \quad \sigma(u) = \frac{1}{1+e^{-u}}\,,\; g(u,v) = e^{-(u^2+v)^3} + u^2 + v^2\,,$$

$b = 1.18$ and $\lambda = 0.05$; the parameters have been chosen to get a challenging but feasible problem. The Bayes classifier is $f^* : x \mapsto \mathbb{1}_{g(x) \geq b}$ and the Bayes risk, computed numerically using the scipy.integrate python library, is approximately equal to 0.242. Agghoo and CV are used with the collection $\mathcal{G}^{\mathrm{NN}}$ with training sets $T_1, \dots, T_V$ that are chosen independently and uniformly among the subsets of $\{1, \dots, n\}$ with cardinality $\lfloor \tau n \rfloor$, for different values of τ and V; hence, CV corresponds to what is usually called "Monte-Carlo CV" [Arlot and Celisse, 2010]. Each algorithm is run on 1000 independent samples of size $n = 500$, and independent test samples of size 1000 are used for estimating the 0–1 excess risks $\ell(f^*, \widehat{f}^{\mathrm{ag}}_{\mathcal{G}^{\mathrm{NN}},\mathcal{T}})$, $\ell(f^*, \widehat{f}^{\mathrm{cv}}_{\mathcal{G}^{\mathrm{NN}},\mathcal{T}})$ and the oracle excess risk $\inf_{G \in \mathcal{G}^{\mathrm{NN}}} \ell(f^*, G(D_n))$. Expectations of these quantities are estimated by taking an average over the 1000 samples; we also compute standard deviations for these estimates, which are not shown on Figure 1 since they are all smaller than 3.6% of the estimated value, so that all visible differences on the graph are significant.

Results are shown on Figure 1. The performance of Agghoo strongly depends on both parameters τ and V. Increasing V improves significantly the performance of the resulting estimator. However, most of the improvement seems to occur between $V = 2$ and $V = 10$, so that taking V much larger seems useless, a behaviour previously observed for CV [Arlot and Lerasle, 2016]. As a function of τ, the performance of Agghoo is U-shaped although not symmetric around $1/2$. It seems that $\tau \in [0.5, 0.8]$ yields better performances, while taking τ close to 0 or 1 should be avoided (at least for $V \leq 20$). Taking V large enough, say $V = 10$, makes the choice of τ less crucial: a large region of values of τ yield (almost) optimal performance. Similar conclusions can be drawn for CV, with the major difference that its performance depends much less on V: only $V = 2$ appears to be significantly worse than $V \geq 5$.

Let us now compare Agghoo with the hold-out (that is, $V = 1$) and CV. For a given τ, Agghoo and CV are much better than the hold-out; there is no surprise here, considering several data splits is always useful. For fixed (τ, V), Agghoo does significantly better than CV if $V \geq 10$, worse if $V = 2$, and they yield similar performance for $V = 5$. Overall, if we can afford the computational cost of $V = 10$ data splits, Agghoo with optimized parameters ($V = 10$, $\tau \in [0.5, 0.8]$) clearly improves over CV with optimized parameters ($V = 10$, $\tau = 0.7$). This advocates for the use of Agghoo instead of CV, unless we have to take $V < 5$ for computational reasons.

4.2 CART decision trees

In a second experiment, we consider the CART classification trees of Breiman et al. [1984].

Experimental setup Data samples of size $n = 500$ are generated, following an experiment of Genuer et al. [2010, Section 2]: $Y \sim \mathcal{B}(0.5)$ and $\varepsilon \sim \mathcal{B}(0.7)$ are independent Bernoulli variables, $\mathcal{X} = \mathbb{R}^d$ with $d \geq 6$ and, given (Y, ε), the coordinates $X^{(j)}$, $j = 1, \ldots, d$ of X are independent with the following distribution. When $\varepsilon = 1$, $X^{(j)} \sim \mathcal{N}(jY, 1)$ for $j \in \{1, 2, 3\}$ and $X^{(j)} \sim \mathcal{N}(0, 1)$ otherwise. When $\varepsilon = 1$, $X^{(j)} \sim \mathcal{N}((j-3)Y, 1)$ for $j \in \{4, 5, 6\}$ and $X^{(j)} \sim \mathcal{N}(0, 1)$ otherwise. We estimate numerically that the Bayes risk is 0.041.

Agghoo and 10-fold CV are used with the family $(G_\alpha^{\mathrm{CART}})_{\alpha>0}$ of pruned CART classifiers, with training sets T_j chosen as in Section 4.1. Let us detail how G_α^{CART} is defined. For any $\alpha > 0$ and any sample D_n, the α-pruned CART classifier $G_\alpha^{\mathrm{CART}}(D_n)$ is defined as the partitioning classifier built on the tree

$$\widehat{t}_{\alpha,n} = \operatorname*{argmin}_{t \subset \widehat{t}} \left\{ \widehat{\mathcal{R}}(\widehat{f}_t) + \alpha |\mathcal{L}_t| \right\} \ ,$$

where $\widehat{t}$ denotes the fully grown CART tree, $\widehat{f}_t$ is the partitioning classifier associated with any tree t, $\widehat{\mathcal{R}} = \widehat{\mathcal{R}}_{\{1,\ldots,n\}}$ is the empirical risk on the whole data set D_n and $|\mathcal{L}_t|$ is the number of leaves (terminal nodes) of the tree t. We also consider random forests (RF), which are a natural competitor to Agghoo applied on pruned CART trees, using the R package `randomForest` [Liaw and Wiener, 2002] with default values for all parameters. Since RF typically build hundred of trees —500 by default—, whereas Agghoo only requires to build around $V = 10$ trees, we also consider RF with `ntree` $= 10$ trees (and default values for all other parameters). The rest of the experimental protocole is the same as in Section 4.1, in particular we consider 1000 samples of size $n = 500$, and risks are estimated with test samples of size 1000. Standard deviations of our risk estimations are not shown since they never exceed 2.5% of the excess risk.

Results are shown on Figure 2, with $d = 7$ (only one noise variable) and $d = 50$ (large majority of noise variables). Concerning Agghoo and CV, the conclusions are similar to the k-NN case, studied in Section 4.1. This is why we only report here their performances for $V = 10$. Let us only notice that Agghoo with $\tau = 0.8$ and $V = 10$ performs almost as well as the oracle, which is a strong competitor since it uses the knowledge of data distribution P. The main novelty here is the comparison to RF: RF with 500 trees clearly outperform all other procedures (and the oracle!), whereas RF with 10 trees works well only for $d = 7$. We explain this fact by noticing that RF builds trees with more randomization than Agghoo (especially when many variables are pure noise, in the $d = 50$ case), so that individual trees (and small forests) perform worse than hold-out selected classifiers, whereas a large enough forest performs much better. Therefore, when such a specific algorithm is available, our advice is to

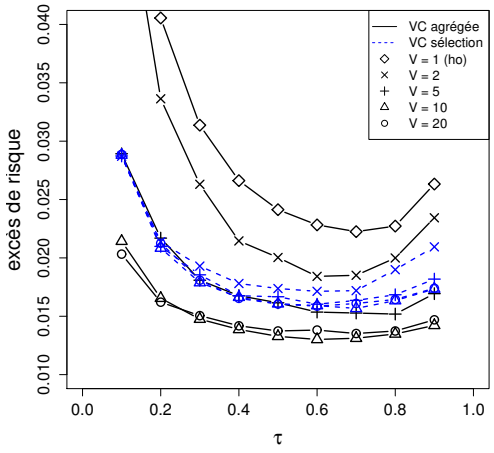

Figure 1: Classification performance of Agghoo and CV for the k-NN family; the oracle excess risk is 0.0034 ± 0.0004

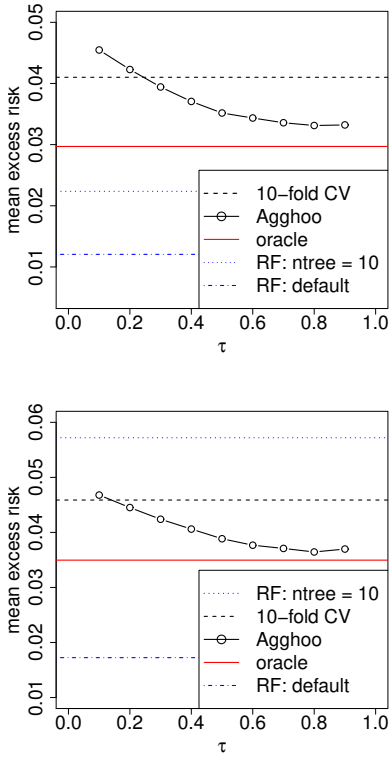

Figure 2: Synthetic dataset (top: $d = 7$, bottom: $d = 50$); classification performance of Agghoo ($V = 10$) and 10-fold CV built on the CART family, as well as the oracle choice and RF

Table 1: Results for the breast-cancer Wisconsin dataset, CART-based classifiers.

Procedure	0–1 risk in %
10-fold CV	6.66 ± 0.06
Agghoo ($V = 10$, $\tau = 0.8$)	5.36 ± 0.05
Oracle	5.08 ± 0.04
RF (`ntree` $= 10$)	3.68 ± 0.04
RF (`ntree` $= 500$)	3.01 ± 0.03

prefer it to Agghoo; but in the general case, it might be difficult to add some randomization individual classifiers, so that Agghoo remains useful. Agghoo can also be interesting when we cannot afford to aggregate more than a few tens of classifiers.

Real-data experiment We also consider the same procedures on the breast-cancer Wisconsin dataset from UCI Machine Learning Repository [Lichman, 2013], for which $n = 699$ and $d = 10$. We use samples of size 500 and the remaining 199 data as test samples. The split between sample and test sample is done 1000 times, so that we can provide error bars for the estimated risks reported in Table 1. The conclusion is similar to what has been obtained on synthetic data with $d = 7$, which confirms our interpretation: when most variables are relevant, RF's individual trees are less randomized so that RF can perform well even with a small number of trees.

5 Discussion

The theoretical and numerical results of the paper show that Agghoo can be used safely in supervised classification with the 0–1 loss, at least when its parameters are properly chosen —$V \geq 10$ and $\tau \in [0.5, 0.8]$ seem to be safe choices. On the theoretical side, Agghoo performs at least as well as the hold-out and it satisfies some sharp oracle inequality. Experiments show that Agghoo actually performs much better, improving over cross-validation except when the number of data splits is strictly smaller than 5, especially when basis classifiers are unstable —like decision trees. Proving theoretically that Agghoo can improve over CV is an exciting, but challenging, open problem that we would like to address in future works.

So, for the same computational cost, Agghoo —with properly chosen parameters V, τ— should be preferred to CV, unless the final classifier has to be interpretable (which makes selection better than aggregation). Yet, Agghoo and CV only are general-purpose tools. For some specific families of classifiers, better procedures can be used. For instance, when one wants a classifier built upon

decision trees, random forests —with sufficiently many trees if there are many irrelevant variables— certainly are a better choice, as shown by our experiments.

Our results can be extended in several ways. First, our theoretical bounds directly apply to subagging hold-out, since it also makes a majority vote among hold-out selected estimators. The difference is that in subagging the training set size is $n-p-q$ and the validation set size is q, for some $q \in \{1, \dots, n-p-1\}$, leading to slightly worse bounds than the ones we obtained for Agghoo (not much if q is well chosen). Oracle inequalities and minimax results can also be obtained for Agghoo in other settings; this paper focuses on two such results for length reasons only. Second, Agghoo can be extended to other learning problems, such as regression and density estimation, replacing the majority vote by an average. Proposition 3.1 then holds true with M replaced by 1 if the loss is convex —for instance, for the least-squares loss—, by Jensen's inequality. Then, Theorem 3.2 could be proved based upon the general results of Massart [2007, Section 8.5]. Third, Agghoo might be improved by weighting the votes of the different hold-out classifiers, as suggested by Jung [2016].

References

S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

S. Arlot and M. Lerasle. Choice of V for V-fold cross-validation in least-squares density estimation. *Journal of Machine Learning Research (JMLR)*, 17(208):1–50, 2016.

J.-Y. Audibert. Fast learning rates in statistical inference through aggregation. *The Annals of Statistics*, 37(4):1591–1646, 2009.

J.-Y. Audibert and A. Tsybakov. Fast learning rates for plug-in classifiers. *Annals of Statistics*, 35(2):608–633, 2007.

G. Biau and L. Devroye. *Lectures on the Nearest Neighbor Method*. Springer Series in the Data Sciences. Springer, 2015.

G. Biau and E. Scornet. A random forest guided tour. *TEST*, 25(2):197–227, 2016.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth Statistics/Probability Series. Wadsworth Advanced Books and Software, Belmont, CA, 1984.

P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data*. Springer Series in Statistics. Springer, Heidelberg, 2011. Methods, theory and applications.

P. Bühlmann and B. Yu. Analyzing bagging. *The Annals of Statistics*, 30(4):927–961, 2002.

K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference*. Springer-Verlag, New York, second edition, 2002. A practical information-theoretic approach.

L. P. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 1996.

T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1, part 2):119–139, 1997. EuroCOLT '95.

R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236, 2010.

Y. Jung. Efficient tuning parameter selection by cross-validated score in high dimensional models. *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, 10(1):19 – 25, 2016.

Y. Jung and J. Hu. A K-fold averaging cross-validation procedure. *Journal of Nonparametric Statistics*, 27(2):167–179, 2015.

G. Lecué. Optimal rates of aggregation in classification under low noise assumption. *Bernoulli*, 13(4):1000–1022, 2007.

A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3): 18–22, 2002. URL `http://CRAN.R-project.org/doc/Rnews/`.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

E. Mammen and A. B. Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.

P. Massart. *Concentration Inequalities and Model Selection*, volume 1896 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007. Lectures from the 33rd Summer School on Probability Theory held in Saint-Flour, July 6–23, 2003, With a foreword by Jean Picard.

A. Nemirovski. *Topics in Non-parametric Statistics*, volume 1738 of *Lecture Notes in Math.* Springer, Berlin, 2000.

B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

I. Steinwart and A. Christmann. *Support vector machines*. Information Science and Statistics. Springer, New York, 2008.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. Methodological*, 58(1):267–288, 1996.

A. B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

Y. Yang. Adaptive regression by mixing. *Journal of the American Statistical Association*, 96(454):574–588, 2001.

A Proof of Proposition 3.1

The proof immediately follows from the following convexity-type property of the majority vote, applied to $(\widehat{f}^{\,\mathrm{ho}}_{\mathcal{G},T_j})_{1\le j\le V}$, since $\mathbb{E}\big[\ell(f^*,\widehat{f}^{\,\mathrm{ho}}_{\mathcal{G},T_j})\big]$ and $\mathbb{E}\big[R(\widehat{f}^{\,\mathrm{ho}}_{\mathcal{G},T_j})\big]$ do not depend on j under assumptions (H1)–(H2) (they only depend on T_j through its cardinality $n-p$).

Proposition A.1 *Let $(\widehat{f}_i)_{1\le i\le V}$ denote a finite family of classifiers and let $\widehat{f}^{mv}$ be some majority vote rule: $\forall x\in\mathcal{X}$, $\widehat{f}^{mv}(x)\in \mathrm{argmax}_{m\in\mathcal{Y}}|\{i\in[V]:\widehat{f}_i(x)=m\}|$. Then,*

$$\ell(f^*,\widehat{f}^{mv})\le\frac{M}{V}\sum_{i=1}^{V}\ell(f^*,\widehat{f}_i)\qquad\text{and}\qquad R(\widehat{f}^{mv})\le\frac{2}{V}\sum_{i=1}^{V}R(\widehat{f}_i)\ .$$

Proof For any $y\in\mathcal{Y}$, define $\eta_y: x\to\mathbb{P}[Y=y|X=x]$. Then, for any $f\in\mathcal{F}$, $R(f)=\mathbb{E}[1-\eta_{f(X)}(X)]$ hence $f^*(X)\in\mathrm{argmax}_{y\in\mathcal{Y}}\eta_y(X)$ and

$$\ell(f^*,f)=\mathbb{E}\Big[\max_{y\in\mathcal{Y}}\eta_y(X)-\eta_{f(X)}(X)\Big]=\mathbb{E}\big[\eta_{f^*(X)}(X)-\eta_{f(X)}(X)\big]\ .$$

We now fix some $x\in\mathcal{X}$ and define $\mathcal{C}_x(y)=\{i\in[V]:\widehat{f}_i(x)=y\}$ and $C_x=\max_{y\in\mathcal{Y}}|\mathcal{C}_x(y)|$. Since $C_xM\ge\sum_{y\in\mathcal{Y}}|\mathcal{C}_x(y)|=V$, it holds $C_x\ge V/M$. On the other hand, by definition of $\widehat{f}^{\mathrm{mv}}$,

$$\frac{1}{V}\sum_{i=1}^{V}\underbrace{\big[\eta_{f^*(x)}(x)-\eta_{\widehat{f}_i(x)}(x)\big]}_{\ge 0}\ge\frac{C_x}{V}\big(\eta_{f^*(x)}(x)-\eta_{\widehat{f}^{\mathrm{mv}}(x)}(x)\big)\ge\frac{1}{M}\big(\eta_{f^*(x)}(x)-\eta_{\widehat{f}^{\mathrm{mv}}(x)}(x)\big)\ .$$

Integrating over x (with respect to the distribution of X) yields the first bound.

For the second bound, fix $x\in\mathcal{X}$ and define $\mathcal{C}_x(y)$ and C_x as above. Let $y\in\mathcal{Y}$ be such that $\widehat{f}^{\mathrm{mv}}(x)\ne y$. Since y occurs less often than $\widehat{f}^{\mathrm{mv}}(x)$ among $\widehat{f}_1(x),\dots,\widehat{f}_V(x)$, we have $|\mathcal{C}_x(y)|\le V/2$. Therefore,

$$\frac{1}{V}\sum_{i=1}^{V}\mathbb{1}_{\{\widehat{f}_i(x)\ne y\}}=\frac{V-|\mathcal{C}_x(y)|}{V}\ge\frac{1}{2}\ .$$

Thus

$$\widehat{f}^{\mathrm{mv}}(x)\ne y\implies\frac{1}{V}\sum_{i=1}^{V}\mathbb{1}_{\{\widehat{f}_i(x)\ne y\}}\ge\frac{1}{2}\ .$$

Hence, for any $y\in\mathcal{Y}$,

$$\mathbb{1}_{\{\widehat{f}^{\mathrm{mv}}(x)\ne y\}}\le\frac{2}{V}\sum_{i=1}^{V}\mathbb{1}_{\{\widehat{f}_i(x)\ne y\}}\ .$$

Taking expectations with respect to (x,y) yields $R(\widehat{f}^{\mathrm{mv}})\le 2V^{-1}\sum_{i=1}^{V}R(\widehat{f}_i)$. ■

B Proof of Theorem 3.2

The proof relies on a result by Massart [2007, Eq. (8.60), which is itself a consequence of Corollary 8.8], which holds true as soon as

$$\forall f \in \mathcal{F}, \qquad \mathrm{Var}\big(\mathbb{1}_{\{f(X)\neq Y\}} - \mathbb{1}_{\{f^*(X)\neq Y\}}\big) \leq \Big[w\big(\sqrt{\ell(f^*,f)}\big)\Big]^2 \tag{4}$$

for some nonnegative and nondecreasing continuous function w on $\mathbb{R}^+$, such that $x \mapsto w(x)/x$ is nonincreasing on $(0,+\infty)$ and $w(1) \geq 1$.

Let us first prove that assumption (4) holds true. On one hand, for any $f \in \mathcal{F}$,

$$\begin{aligned}\mathrm{Var}\big(\mathbb{1}_{\{f(X)\neq Y\}} - \mathbb{1}_{\{f^*(X)\neq Y\}}\big) &\leq \mathbb{E}[|\mathbb{1}_{\{f(X)\neq Y\}} - \mathbb{1}_{\{f^*(X)\neq Y\}}|^2] \\ &= \mathbb{E}[\mathbb{1}_{\{f(X)\neq f^*(X)\}}] = \mathbb{E}\big[|f(X) - f^*(X)|\big] \ . \end{aligned} \tag{5}$$

On the other hand, since we consider binary classification with the 0–1 loss, for any $f \in \mathcal{F}$ and $h > 0$,

$$\begin{aligned}\ell(f^*,f) &= \mathbb{E}\big[|2\eta(X)-1| \cdot |f(X) - f^*(X)|\big] && \text{by Devroye et al. [1996, Theorem 2.2]} \\ &\geq h\mathbb{E}\big[|f(X) - f^*(X)|\mathbb{1}_{\{|2\eta(X)-1|\geq h\}}\big] \\ &\geq h\mathbb{E}\big[|f(X) - f^*(X)| - \mathbb{1}_{\{|2\eta(X)-1|< h\}}\big] && \text{since } |f-f^*| \leq 1 \\ &\geq h\mathbb{E}\big[|f(X) - f^*(X)|\big] - ch^{\beta+1} && \text{by (MA).}\end{aligned}$$

This lower bound is maximized by taking

$$h = h_* := \left(\frac{\mathbb{E}\big[|f(X) - f^*(X)|\big]}{c(\beta+1)}\right)^{\frac{1}{\beta}} \ ,$$

which belongs to $[0,1]$ since $c \geq 1$ and $\mathbb{E}\big[|f(X) - f^*(X)|\big] \leq 1$. Thus, we obtain

$$\ell(f^*,f) \geq h_* \frac{\beta}{\beta+1}\mathbb{E}\big[|f(X) - f^*(X)|\big] = \frac{\beta}{(\beta+1)^{(\beta+1)/\beta}c^{1/\beta}}\mathbb{E}\big[|f(X) - f^*(X)|\big]^{(\beta+1)/\beta}$$

hence Eq. (5) leads to

$$\mathrm{Var}\big(\mathbb{1}_{\{f(X)\neq Y\}} - \mathbb{1}_{\{f^*(X)\neq Y\}}\big) \leq \mathbb{E}\big[|f(X) - f^*(X)|\big] \leq \frac{\beta+1}{\beta^{\beta/(\beta+1)}}c^{\frac{1}{\beta+1}}\ell(f^*,f)^{\frac{\beta}{\beta+1}} \leq 2c^{\frac{1}{\beta+1}}\ell(f^*,f)^{\frac{\beta}{\beta+1}} \ .$$

Therefore, Eq. (4) holds true with $w(u) = \sqrt{c_1}u^{\frac{\beta}{\beta+1}}$ and $c_1 = 2c^{\frac{1}{\beta+1}}$, which statisfies the required conditions. So, by Massart [2007, Eq. (8.60)], for any $\theta \in (0,1)$,

$$\mathbb{E}\big[\ell(f^*, \widehat{f}^{\,\mathrm{ho}}_{\mathcal{G},T}) \,|\, D_n^T\big] \leq \frac{1+\theta}{1-\theta}\inf_{G\in\mathcal{G}}\ell(f^*,\widehat{f}_{G,T}) + \frac{\delta_*^2}{1-\theta}\left[2\theta + \log(e|\mathcal{G}|)\left(\frac{1}{3}+\theta^{-1}\right)\right] \tag{6}$$

where δ_* is the positive solution of the fixed-point equation $w(\delta_*) = \sqrt{p}\delta_*^2$, that is $\delta_*^2 = (c_1/p)^{\frac{\beta+1}{\beta+2}}$. Taking expectations with respect to the training data D_n^T, we obtain

$$\mathbb{E}\big[\ell(f^*, \widehat{f}^{\text{ho}}_{\mathcal{G},T})\big] \leq \frac{1+\theta}{1-\theta}\mathbb{E}\left[\inf_{G\in\mathcal{G}} \ell(f^*, \widehat{f}_{G,T})\right] + \frac{2c^{\frac{1}{\beta+2}}}{1-\theta}\frac{2\theta + \log(e|\mathcal{G}|)\left(\frac{1}{3}+\theta^{-1}\right)}{p^{\frac{\beta+1}{\beta+2}}} \ .$$

Now, by Proposition 3.1,

$$\mathbb{E}\big[\ell(f^*, \widehat{f}^{\text{ag}}_{\mathcal{G},T})\big] \leq 2\mathbb{E}\big[\ell(f^*, \widehat{f}^{\text{ho}}_{\mathcal{G},T_1})\big] \leq 2\frac{1+\theta}{1-\theta}\mathbb{E}\left[\inf_{G\in\mathcal{G}} \ell(f^*, \widehat{f}_{G,T})\right] + \frac{4c^{\frac{1}{\beta+2}}}{1-\theta}\frac{2\theta + \log(e|\mathcal{G}|)\left(\frac{1}{3}+\theta^{-1}\right)}{p^{\frac{\beta+1}{\beta+2}}} \ .$$

Taking $\theta = 1/5$ leads to the result. ■

C Proofs of minimax results

C.1 Formal framework and result

To begin with, let us recall the definition of minimax optimality.

Definition C.1 *Let $\mathcal{P}$ be a class of probability distributions over $\mathcal{X}\times\mathcal{Y}$. A classifier $\widehat{f}_n$ is said to be minimax over the class $\mathcal{P}$ when its maximal excess loss on $\mathcal{P}$ satisfies*

$$\sup_{P\in\mathcal{P}} \mathbb{E}_P\left[\ell(f_P^*, \widehat{f}_n)\right] \leq \kappa(\mathcal{P}) \inf_G \sup_{P\in\mathcal{P}} \mathbb{E}_P\left[\ell(f_P^*, G(D_n)))\right] \ .$$

Any sequence u_n such that $u_n^{-1}\inf_G \sup_{P\in\mathcal{P}} \mathbb{E}_P\left[\ell(f_P^*, G(D_n))\right]$ is bounded and bounded away from 0 is called a minimax rate of convergence.

We now define formally $\Sigma_{\gamma,L}$.

Definition C.2 *For any $k = (k_1, ..., k_d) \in \mathbb{N}^d$, any $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and any sufficiently smooth function $f : \mathbb{R}^d \to \mathbb{R}$, let*

$$|k| = \sum_{i=1}^d k_i, \quad k! = \prod_{i=1}^d k_i!, \quad D^k f(x) = \frac{\partial^{|k|}}{\partial x_1^{k_1}...\partial x_d^{k_d}} f(x_1, ..., x_d), \quad x^k = \prod_{i=1}^d x_i^{k_i} \ .$$

For any $\alpha \in \mathbb{N}$, any $\mathcal{C}^\alpha$-function f and any point $x \in \mathbb{R}^d$, let

$$Q_x^\alpha f(y) = \sum_{k:|k|\leq\alpha} \frac{D^k f(x)}{k!}(y-x)^k \ .$$

The class of γ-smooth functions with constant L is defined as

$$\Sigma_{\gamma,L}(\mathbb{R}^d) = \left\{ f \in \mathcal{C}^{\lfloor\gamma\rfloor}\left(\mathbb{R}^d, \mathbb{R}\right) : \quad \forall x, y \in \mathbb{R}^d, \quad |f(y) - Q_x^{\lfloor\gamma\rfloor}(y)| \leq L\left\|y-x\right\|^\gamma \right\} \ .$$

Let us conclude this section with the definition of the minimax classification rules.

Estimators achieving the minimax rates over the classes $\mathcal{P}_{\gamma,L,\beta,c}$ were obtained by Audibert and Tsybakov [2007]. Let us recall this construction.

Definition C.3 *Let $\ell \in \mathbb{N}$, $h > 0$ and let K denote the standard gaussian Kernel. For any polynomial Q, let*

$$\mathcal{C}_{h,x}(Q) = \sum_{i=1}^{n} (Y_i - Q(X_i - x))^2 K\left(\frac{X_i - x}{h}\right) \ .$$

Denote by

$$\widehat{Q}_{\ell,h,x} = \operatorname*{argmin}_{Q:\deg Q \le \ell} \mathcal{C}_{h,x}(Q)$$

if the minimum exists and is unique and $\widehat{Q}_{\ell,h,x} = 0$ otherwise.

Let also $\bar{B}$ be the matrix $(B_{s_1,s_2})_{|s_1|,|s_2| \le \lfloor\gamma\rfloor}$ where

$$B_{s_1,s_2} = \frac{1}{nh^d} \sum_{i=1}^{n} \left(\frac{X_i - x}{h}\right)^{s_1+s_2} K\left(\frac{X_i - x}{h}\right) \ .$$

Let $\lambda_{min} = \sup\{\lambda \in \mathbb{R} \,|\, \forall u \in \mathbb{R}^d \left\|\bar{B}u\right\| \ge \lambda \|u\|\}$. The local polynomial rule with degree ℓ and bandwidth h is defined by $G^{\mathrm{LP}}_{\ell,h,x}(D_n) : x \mapsto \mathbb{1}_{\{\widehat{\eta}_{\ell,h}(x) \ge \frac{1}{2}\}}$, where

$$\widehat{\eta}_{\ell,h}(x) = \begin{cases} \widehat{Q}_{\ell,h,x}(0) & \text{if } \lambda_{min} \ge (\log n)^{-1} \\ 0 & \text{otherwise} \end{cases} \ .$$

Theorem C.4 *The rate $u_n = n^{-\frac{\gamma(1+\beta)}{2\gamma+d}}$ is minimax over the classes $\mathcal{P}_{\gamma,L,\beta,c}$, for any $\beta \ge 0$, $\gamma > 0$, $L > 0$, $c \ge 1$ and $\gamma\beta < d$.*

Furthermore, if $n^{\frac{1}{2\gamma+d}} h_n \in [\tau, \tau']$ for some $\tau' > \tau > 0$, the estimator $G^{\mathrm{LP}}_{\lfloor\gamma\rfloor,h_n}(D_n)$ is minimax over $\mathcal{P}_{\gamma,L,\beta,c}$ for any $L > 0, c > 0, \gamma\beta < d$.

Proof The lower bound is proved in [Audibert and Tsybakov, 2007, Theorem 3.5].

For the upper bound, we shall denote by C_1, C_2, C_3 functions of the parameters γ, L, a, b, d but neither h nor n which may vary from line to line. Eq (3.7) in [Audibert and Tsybakov, 2007, Theorem 3.2] implies that

$$\sup_{P \in \mathcal{P}_{\gamma,L,\beta,c}} P(|\widehat{\eta}_{\lfloor\gamma\rfloor,h}(x) - \eta(x)| \ge \delta) \le C_1 \exp(-C_2 n h^d \delta^2)$$

for h, δ such that $0 < h \le L^2$ and $C_3 h^\gamma \le \delta$.

Then our choice of h_n yields, for any $\delta \ge C_3 h_n^\gamma$,

$$\sup_{P \in \mathcal{P}_{\gamma,L,\beta,c}} P(|\widehat{\eta}_{\lfloor\gamma\rfloor,h_n}(x) - \eta(x)| \ge \delta) \le C_1 \exp(-C_2 n^{\frac{2\gamma}{2\gamma+d}} \delta^2) \ . \tag{7}$$

Moreover, for $\delta < C_3 h_n^\gamma$,

$$nh_n^d\delta^2 \le C_3 nh_n^{2\gamma+d} \le C_3 b^{2\gamma+d} \ ,$$

therefore, inequality (7) holds for all $\delta > 0$ and [Audibert and Tsybakov, 2007, Lemma 3.1] concludes the proof. ■

C.2 Proof of Theorem 3.4

Let $k_n = \lfloor n^{\frac{1}{2\gamma+d}} \rfloor$ and $h_n = 1/k_n$. Then for $n > d$, the risk of the oracle is bounded from above by

$$\mathbb{E}\left[\inf_{G\in\mathcal{G}_n^{\mathrm{LP}}} \ell(f^*, \widehat{f}_{G,T_1})\right] \le \inf_{G\in\mathcal{G}_n^{\mathrm{LP}}} \mathbb{E}[\ell(f^*, \widehat{f}_{G,T_1})] \le \mathbb{E}\big[\ell(f^*, \widehat{f}_{G^{\mathrm{LP}}_{\lfloor\gamma\rfloor,h_n},T_1})\big] \ .$$

Moreover,

$$1 = \lfloor n^{\frac{1}{2\gamma+d}} \rfloor h_n \le n^{\frac{1}{2\gamma+d}} h_n \le (\lfloor n^{\frac{1}{2\gamma+d}} \rfloor + 1) h_n \le 1 + h_n \le 2 \ .$$

Therefore $1 \le n^{\frac{1}{2\gamma+d}} h_n \le 2$ and, by Theorem C.4 and the fact that $n - p \ge \tau n - 1 \ge \tau n/2$ for $n \ge 2/\tau$, there exist constants $C_\cdot$ depending on various parameters indicated in subscript, but not on n such that

$$\mathbb{E}[\ell(f^*, \widehat{f}_{G^{\mathrm{LP}}_{\lfloor\gamma\rfloor,h_n},T_1})] \le C_{\gamma,L,\beta,c}(n-p_n)^{-\frac{\gamma(1+\beta)}{2\gamma+d}} \le C_{\gamma,L,\beta,c,\tau} n^{-\frac{\gamma(1+\beta)}{2\gamma+d}} \ .$$

By Theorem 3.2, there exists a constant $C_{c,\beta}$ such that

$$\mathbb{E}\big[\ell(f^*, \widehat{f}^{\mathrm{ag}}_{\mathcal{G}_n^{\mathrm{LP}},\mathcal{T}_n})\big] \le C_{c,\beta}\left(\mathbb{E}\big[\ell(f^*, \widehat{f}_{G^{\mathrm{LP}}_{\lfloor\gamma\rfloor,h_n},T_1})\big] + \frac{\log(e|\mathcal{G}_n^{\mathrm{LP}}|)}{p_n^{\frac{\beta+1}{\beta+2}}}\right) \ .$$

Since $p_n \ge (1-\tau)n$ and $|\mathcal{G}_n^{\mathrm{LP}}| = n^2$, it follows that

$$\mathbb{E}\big[\ell(f^*, \widehat{f}^{\mathrm{ag}}_{\mathcal{G}_n^{\mathrm{LP}},\mathcal{T}_n})\big] \le C_{\gamma,L,\beta,c,\tau} n^{-\frac{\gamma(1+\beta)}{2\gamma+d}} + C_{c,\beta}\frac{\log n}{n^{\frac{\beta+1}{\beta+2}}} \ . \tag{8}$$

Finally, since $\gamma\beta < d$, $(\beta+1)/(\beta+2) > [\gamma(1+\beta)]/[2\gamma+d]$, and therefore

$$\frac{\log n}{n^{\frac{\beta+1}{\beta+2}}} \le C_{\beta,\gamma,d} n^{-\frac{\gamma(1+\beta)}{2\gamma+d}} \ .$$

It follows from Eq. (8) that the Agghoo estimator is minimax. ■